AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (currently amended): A programming tool for at least one of creating and displaying programs to control a flow of a process using a graphics language for simultaneous representation in a diagram, on a display device, of ~~a sequence over time and~~ interactions of objects that are involved in the control of the process and of a sequence of the interactions of the objects over time,

wherein a coordination element is provided, which ~~manages the sequence over time and~~ establishes, controls and monitors the interactions of the objects involved and the sequence of the interactions of the objects over time, and

wherein the coordination element detects and defines, based on an interpretation of calls, parameter queries and measurement results, which object should be addressed and actuated.

2. (original): A programming tool as claimed in Claim 1, wherein the process is an automation technology process.

3. (original): A programming tool as claimed in Claim 1, wherein the process is a technical process.

4. (original): A programming tool as claimed in Claim 1, wherein the program is executed in plural, distributed stored program controllers.

5. (original): A programming tool as claimed in Claim 1, wherein a virtual or additional real processor is provided as the coordination element.

6. (original): A programming tool as claimed in Claim 4, wherein a virtual processor or an additional real processor is provided as the coordination element in connection with the distributed stored program controllers.

7. (original): A programming tool as claimed in Claim 1, wherein at least substantially all calls of the objects are processed by the coordination element.

8. (original): A programming tool as claimed in Claim 7, wherein the coordination element determines at least one of the instant of each call and the addressee of each call.

9. (original): A programming tool as claimed in Claim 1, wherein the graphics language comprises a graphic representation of all of the objects and a graphic representation of all of the object interactions, wherein each graphic representation, of the objects and the object interactions, respectively, is called and interconnected using an editor to implement an executable program.

10. (original): A programming tool as claimed in Claim 9, wherein each graphic representation in the diagram of an object and an object interaction is associated with an instruction or a program module.

11. (original): A programming tool as claimed in Claim 10, wherein the instruction or the program module is in machine language.

12. (original): A programming tool as claimed in Claim 9, wherein the following additional object interactions:

branching of an object call;

parallel connection of an object call;

synchronized connection of at least two interactions; or

loop or jump to repeat at least one of an instruction and a program segment;

are each represented conditionally or unconditionally in the diagram and are thereby implemented correspondingly.

13. (original): A programming tool as claimed in Claim 1, wherein a representation of the graphics language in the diagram shows the object interactions a first axis, and shows a sequence of the object interactions over time on a second axis of the diagram.

14. (original): A programming tool as claimed in Claim 13, wherein the representation of the graphics language in the diagram is real-time capable.

15. (original): A programming tool as claimed in Claim 14, wherein the display device is associated with a buffer memory for buffered representation of the flow of the process using the graphics language.

16. (original): A programming tool as claimed in Claim 13, wherein a sequence chart representation is selected as the diagram.

17. (original): A programming tool as claimed in Claim 13, wherein the diagram shows the sequence of object interactions over time on the second axis of the diagram from top to bottom.

18. (original): A programming tool as claimed in Claim 13, wherein the graphics language in the diagram can be constructed in real time.

19. (currently amended): A method for programming and representing a program run for at least one of open-loop and closed-loop control of a process, using at least one programmable controller, in which a graphics language is used to implement a process capable of being represented by objects and object interactions, comprising:

calling a plurality of objects involved in the process in a common diagram;

calling a plurality of respectively required object interactions in the common diagram;

editing the selected objects and object interactions, as well as the sequence of the object interactions over time, in the common diagram;

providing a coordination element that establishes, controls and monitors the interactions of the objects involved and the sequence of the interactions of the objects over time and that detects and defines, based on an interpretation of calls, parameter queries and measurement results, which object should be addressed and actuated; and

translating the previously implemented program into at least one of a corresponding high-level language and a corresponding machine language.

20. (original): A method as claimed in Claim 19, wherein the objects and the object interactions are arranged on a first axis of the common diagram, and wherein the successive sequence of the object interactions over time is represented by arranging the object interactions on a second axis of the common diagram.

21. (original): A method for programming as claimed in claim 19, wherein the process is an automation technology process.

22. (original): A method for programming as claimed in Claim 19, wherein the process is a technical process.

23. (original): A method for programming as claimed in Claim 20, wherein at least one of the arrangement of the objects and the object interactions, and the representation of the successive sequence of the object interactions over time, in the common diagram are real-time capable.

24. (original): A method for programming as claimed in Claim 20, wherein a sequence chart representation is selected as the common diagram.

25. (original): A method for programming as claimed in Claim 20, wherein the common diagram is two-dimensional.

26. (original): A method for programming as claimed in Claim 20, wherein the successive sequence of the object interactions over time is represented by arranging the object interactions from top to bottom on the second axis of the common diagram.

27. (currently amended): A programming tool for creating and providing a graphic representation in a diagram of programs that control the flow of a process, comprising:

a coordination element that ~~manages~~ establishes, controls and monitors the interactions of objects that are involved in the control of the process and ~~manages~~ establishes, controls and monitors a sequence of the object interactions over time and that detects and defines, based on an interpretation of calls, parameter queries and measurement results, which object should be addressed and actuated; and

a display device that provides a graphic representation of the object interactions together with a graphic representation of the sequence of the object interactions over time in the diagram.

28. (original): A programming tool as claimed in Claim 27, wherein the process is an automation technology process.

29. (original): A programming tool as claimed in Claim 27, wherein the process is a technical process.

30. (original): A programming tool as claimed in Claim 27, further comprising plural distributed stored program controllers in which the program is executed.

31. (original): A programming tool as claimed in Claim 27, wherein the coordination element comprises a virtual processor or an additional real processor.

32. (original): A programming tool as claimed in Claim 30, wherein the coordination element comprises a virtual processor or an additional real processor in connection with the plural distributed stored program controllers.

33. (original): A programming tool as claimed in Claim 27, wherein at least substantially all calls of the objects are processed by the coordination element.

34. (original): A programming tool as claimed in 33, wherein the coordination element determines at least one of the instant of each call and the addressee of each call.

35. (original): A programming tool as claimed in Claim 27, further comprising an editor that calls and interconnects the graphic representation of the objects and the graphic representation of the object interactions, respectively, to implement an executable program.

36. (original): A programming tool as claimed in Claim 35, wherein each graphic representation in the diagram of an object and an object interaction is associated with an instruction or a program module.

37. (original): A programming tool as claimed in claim 36, wherein the instruction or the program module is in machine language.

38. (original): A programming tool as claimed in Claim 35, wherein the following additional object interactions:

branching of an object call;

parallel connection of an object call;

synchronized connection of at least two interactions; or

loop or jump to repeat an instruction and/or a program segment;

are each represented conditionally or unconditionally in the diagram and are thereby implemented correspondingly.

39. (original): A programming tool as claimed in Claim 27, wherein the graphic representation in the diagram shows the object interactions on a first axis, and shows a sequence of the object interactions over time on a second axis of the diagram.

40. (original): A programming tool as claimed in Claim 39, wherein the diagram shows the sequence of object interactions over time on the second axis of the diagram from top to bottom.

41. (original): A programming tool as claimed in Claim 39, wherein the graphic representation is depicted in real-time.

42. (original): A programming tool as claimed in Claim 39, wherein the graphic representation is constructed in real-time.

43. (original): A programming tool as claimed in Claim 41, wherein the display device is associated with a buffer memory for buffered representation of the graphic representation.

44. (original): A programming tool as claimed in Claim 39, wherein a sequence chart representation is selected as the diagram.